

# Python Installationshinweise

Edition 11.2022

## Inhalt

<b>1</b>	<b>Übersicht .....</b>	<b>2</b>
<b>2</b>	<b>Kurseinrichtung .....</b>	<b>3</b>
2.1	Miniconda.....	3
2.2	Visual Studio Code.....	4
2.3	Kurs Beispiele .....	6
2.4	Projekt Setup.....	7
2.5	Installation Bibliotheken.....	8
2.6	Installation Jupiter Notebook .....	9
<b>3</b>	<b>Weitere Informationen .....</b>	<b>10</b>
3.1	Ausführung Python Code .....	10
3.2	Ausführung Python Code Snippet .....	10
3.3	Debug Python Code .....	11
3.4	Installation und Verwendung weiterer Python Bibliotheken .....	12
3.5	Aktivierung von Unit Testing im VS Code.....	13
<b>4</b>	<b>Ubuntu Setup.....</b>	<b>14</b>
4.1	Miniconda.....	14
4.2	Visual Studio Code.....	15
4.3	GIT .....	16
4.4	Course Beispiele .....	16
4.5	Project Setup.....	16
<b>5</b>	<b>Anhang .....</b>	<b>17</b>
5.1	VS Code – Tipps .....	17
5.2	VS Code – Terminal Settings .....	17



## 1 Übersicht

Die folgende Anleitung ist nur eine Option, um Ihre Python-Umgebung für den Kurs vorzubereiten. Wenn Sie bereits über eine laufende Python-Umgebung mit einer geeigneten Entwicklungsumgebung verfügen, brauchen Sie nur die Kursbeispiele herunterzuladen und zu prüfen, ob das Beispiel `hello.py` läuft und die Bibliotheken aus Kapitel 2.5 und 2.6 zu installieren.

Falls nicht, folgen Sie bitte den Installationsschritten in **Kapitel 2 Kurseinrichtung** und stellen Sie sicher, dass das **hello.py** Beispiel läuft.

### **Wichtig**

**Sie benötigen Administratoren Rechte für die Installation.**

**Wir können während des Kurses keine Unterstützung bei der Installation leisten.**



## 2 Kurseinrichtung

### 2.1 Miniconda

Miniconda ist ein freie verfügbares minimales Installationsprogramm für conda. Es ist eine kleine Bootstrap-Version von Anaconda, die nur conda, Python, die Pakete, von denen sie abhängen, und eine kleine Anzahl anderer nützlicher Pakete, einschließlich pip, zlib und ein paar andere, enthält.

#### 1 – Download und Installation von Miniconda für Python 3.8 oder höher

<https://docs.conda.io/en/latest/miniconda.html>

Windows installers			
Windows			
Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Windows 64-bit	57.0 MiB	4fa22bba0497babb5b6608cb8843545372a99f5331c8120099ae1
	Miniconda3 Windows 32-bit	54.2 MiB	9c2ef76bae97246c85c206733ca30fd1feb8a4b3f90a2a511fea6

#### 2 – Einstellungen während der Installation:

- 1) Choose an installation location or use the default proposed by Miniconda
- 2) Enable Checkbox: Register Miniconda as my default Python
- 3) Finish Setup



## 2.2 Visual Studio Code

Installation und Konfiguration von Microsoft Visual Studio Code.

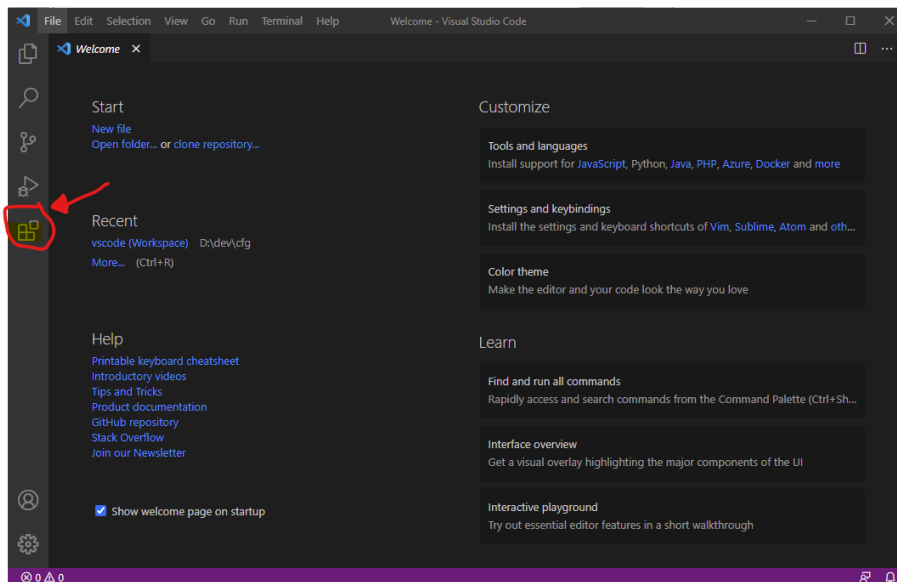
### 1 - Download und Installation Visual Studio Code:

<https://code.visualstudio.com>

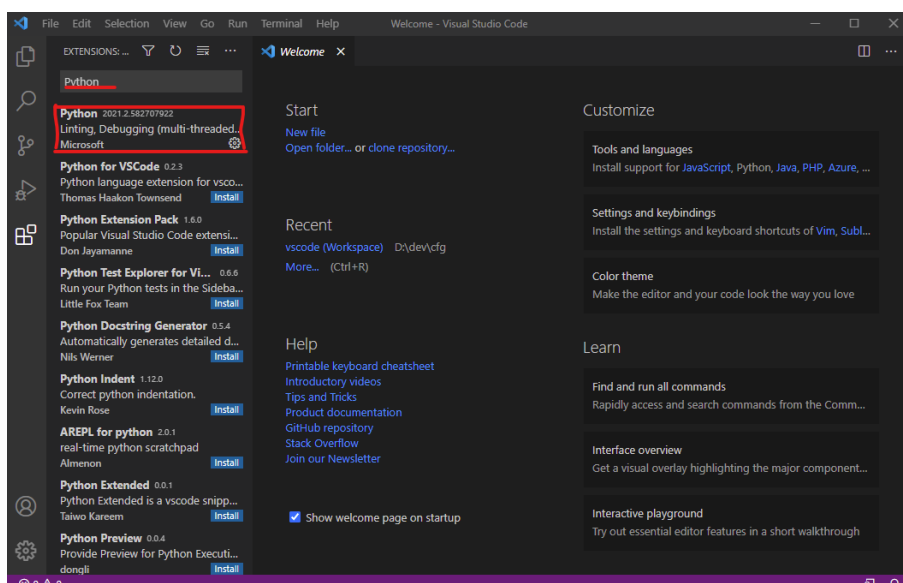
### 2 - Start VS Code und Installation der Python Extension

### 3 – Installation der Python Extension

Auf der linken Seite, den Button «Plugin» wählen:



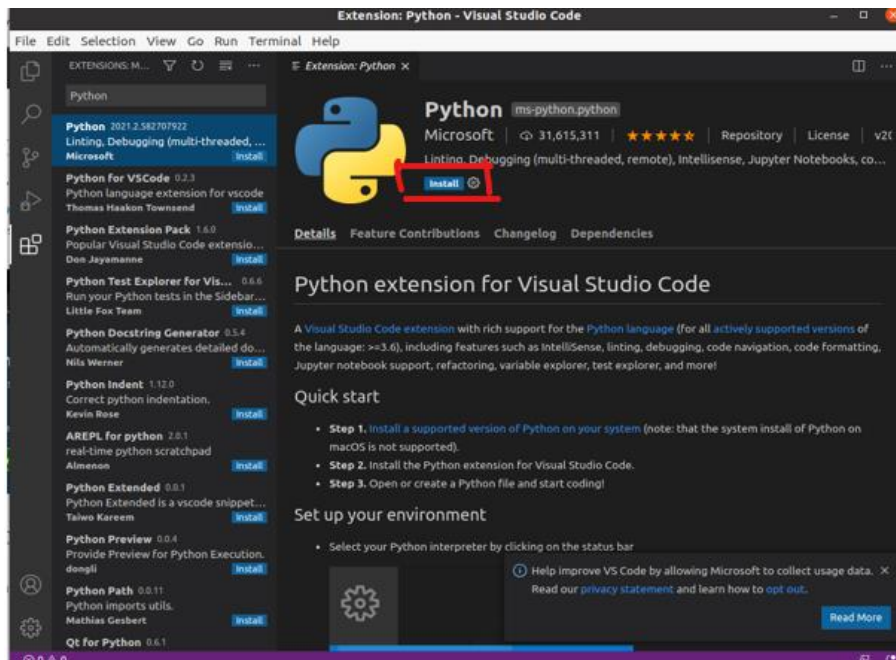
Im Suchfeld nach «Python» suchen



# Python Installationshinweise



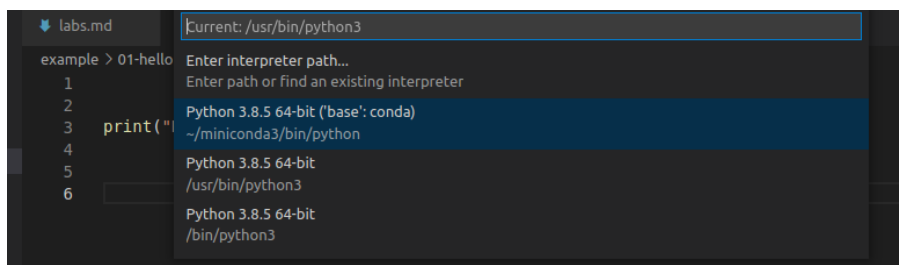
Auswahl des **Python** Plugin und anschliessend **Install** Button wählen



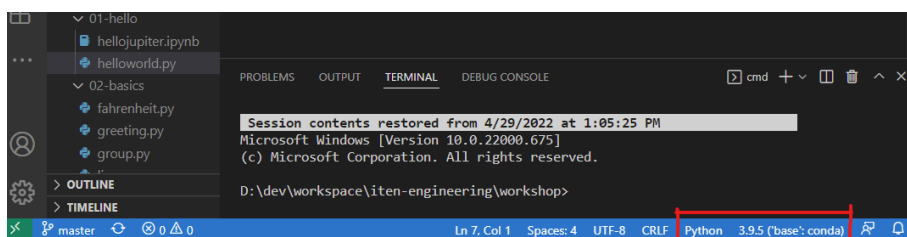
Beachte: Dies installiert auch das Jupiter Plugin.

## 4 – Auswahl eines Python Interpreter:

1. Button drücken: Ctrl+Shift+P
2. Eingabe: Python: Select Interpreter
3. Auswahl: ~/miniconda... Installation



Sobald eine Python Datei angewählt ist, sieht man die Python Interpreter Einstellung unter rechts im VS Code:



## Weitere Informationen

Siehe: <https://code.visualstudio.com/docs/python/python-tutorial>



### 2.3 Kurs Beispiele

Für den Kurs benötigen wir das folgende GitHub Projekt:

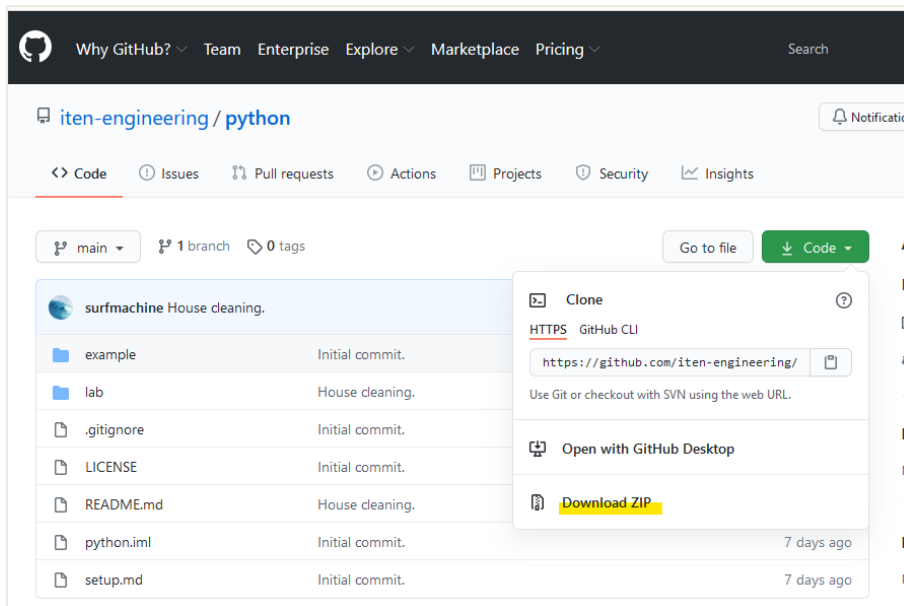
- <https://github.com/iten-engineering/python>

Das Projekt enthält die Slices, Beispiele sowie alle Übungen, die wir während des Kurses durchführen. Das Projekt müssen wir jetzt installieren.

Sie können entweder eine ZIP-Datei herunterladen oder, wenn Sie GIT installiert haben, das Projekt mit git clone installieren.

#### Option A – Download Kurs Beispiele als ZIP

<https://github.com/iten-engineering/python>



Laden Sie die ZIP-Datei herunter und entpacken Sie den Inhalt in Ihr Arbeitsverzeichnis.  
Zum Beispiel `c:\course\workspace\python`

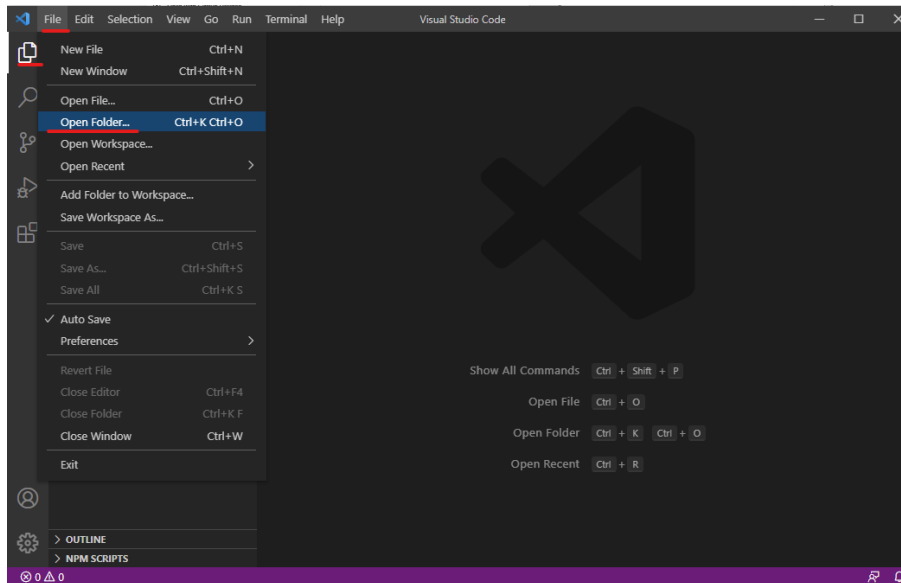
#### Option B – Installation mit GIT

- Erstellen Sie ein Workspace-Verzeichnis für das Kursprojekte.
- Öffnen Sie die Git Bash und wechseln Sie in das Workspace-Verzeichnis
- Führen Sie den folgenden Befehl aus:  
`$> git clone https://github.com/iten-engineering/python`



## 2.4 Projekt Setup

Start VS Code, Click Explorer Button oben links, dann wähle: File / Open Folder:

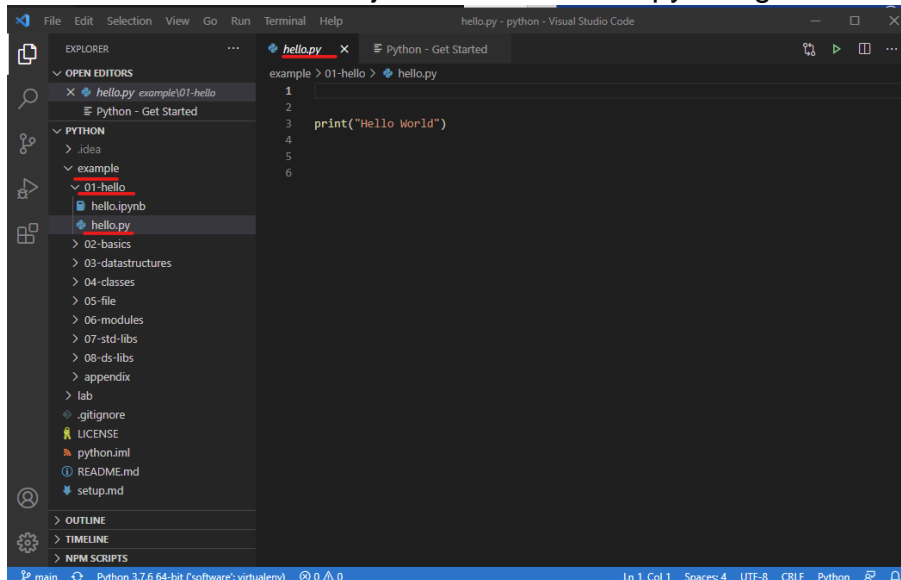


Navigieren Sie zum Kursordner und wählen sie den Ordner.

Zum Beispiel: `c:\course\workspace\python`

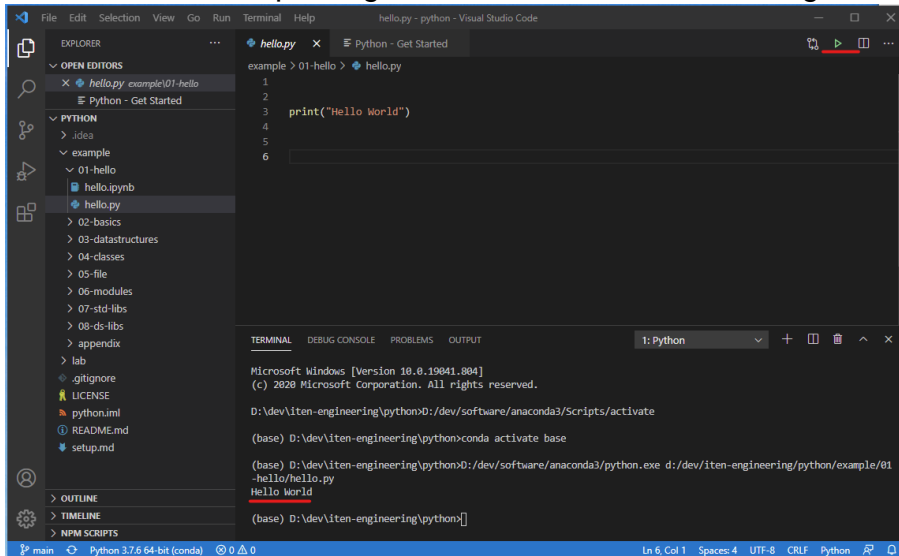
Jetzt wird das Projektverzeichnis geladen und die Ordner und Dateien sind sichtbar.

Auf der linken Seite kann jetzt zur Datei `hello.py` navigiert werden. Klicken Sie diese an:





Jetzt kann das Script ausgeführt werden. Klicke die den grünen Pfeil oben rechts:



Wenn Ihre Ausgabe ähnlich aussieht wie der Screenshot, ist alles in Ordnung.  
Herzlichen Glückwunsch zu Ihrem ersten Hello World-Beispiel mit Python!

Wenn Ihre Ausgabe die folgende Fehlermeldung anzeigt, befolgen Sie bitte die Schritte im Anhang 5.2 "VS-Code - Terminaleinstellung" und versuchen Sie es erneut.

conda : The term 'conda' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again. At line:1 char:1

## 2.5 Installation Bibliotheken

Für die fortgeschrittenen Beispiele und Showcases benötigen wir einige weitere Bibliotheken. Um alles bereit zu haben, installieren Sie bitte die folgenden Bibliotheken.

Öffnen Sie dazu ein Terminal in VS Code (oder die Anaconda-Shell) und führen Sie die folgenden Befehle aus:

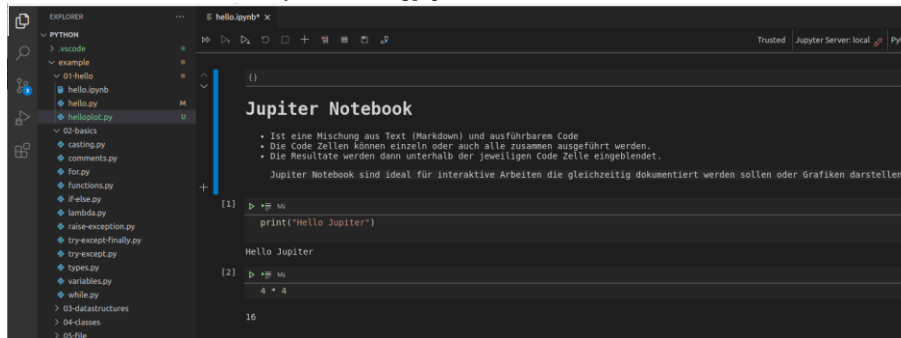
```
pip install matplotlib
pip install openpyxl
pip install Flask
pip install -U scikit-learn
pip install streamlit
```



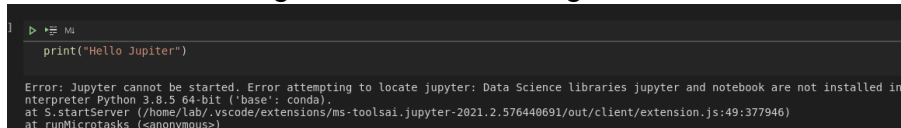


## 2.6 Installation Jupiter Notebook

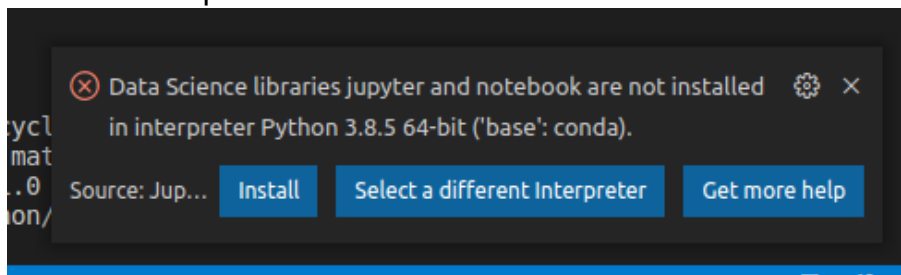
### 1. Öffnen Sie das Script `hello.jpynb`:



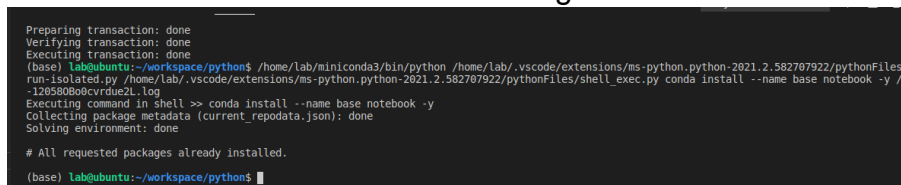
### 2. Wenn Sie die erste Zelle zum ersten Mal ausführen, erhalten Sie die folgende Fehlermeldung:



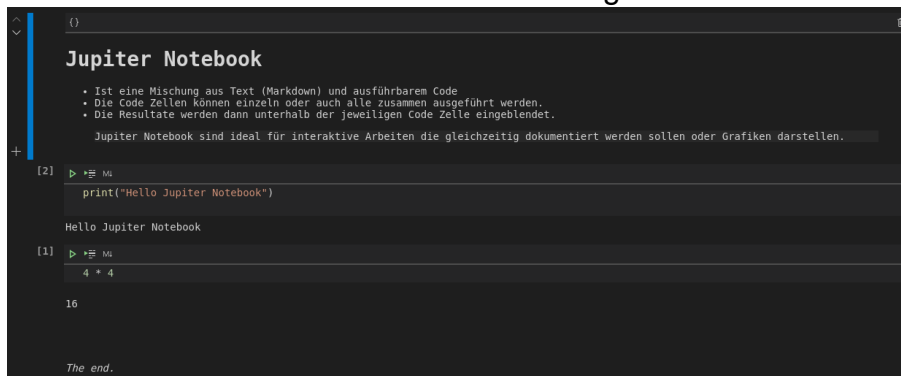
### 3. Installation Jupiter: Click "Install" Button:



Anschließend wird die Installation ausgeführt:



### 4. Jetzt können die Zellen des Notebook ausgeführt werden:





### 3 Weitere Informationen

Dieses Kapitel enthält weitere Informationen, die wir während dem Kurs verwendet werden. Sie müssen diese jetzt nicht zwingend als Teil der Kursvorbereitung ausführen. Wenn Sie aber noch Zeit und Lust haben, können Sie gerne weitermachen.

#### 3.1 Ausführung Python Code

Erstellen Sie eine neue Datei **hello.py** mit dem folgenden Inhalt:

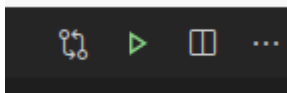
```

1
2
3
4 msg = "Hello World"
5 print(msg)
6

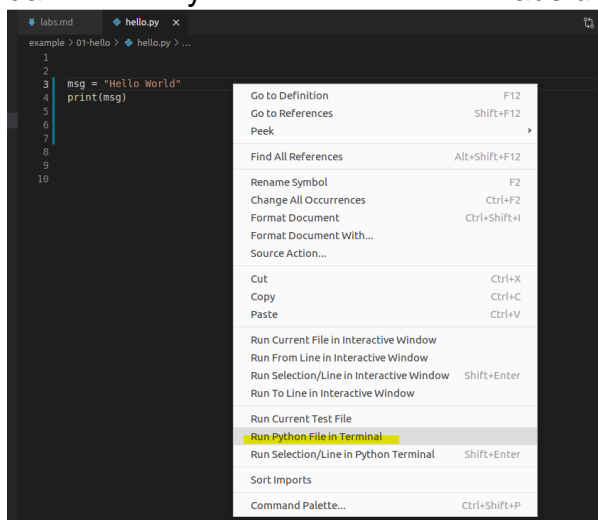
```

Ausführung des Python Scripts:

1. Drücken Sie den Grünen Pfeil Button oben rechts:



2. Alternativ können sie die rechte Maustaste wählen (in der offenen Datei) und dann "Run Python File in Terminal" ausführen.



#### 3.2 Ausführung Python Code Snippet

Markieren Sie einen Teil der Befehle im Python Script.

Drücken Sie SHIFT + ENTER um die markierten Befehle auszuführen.



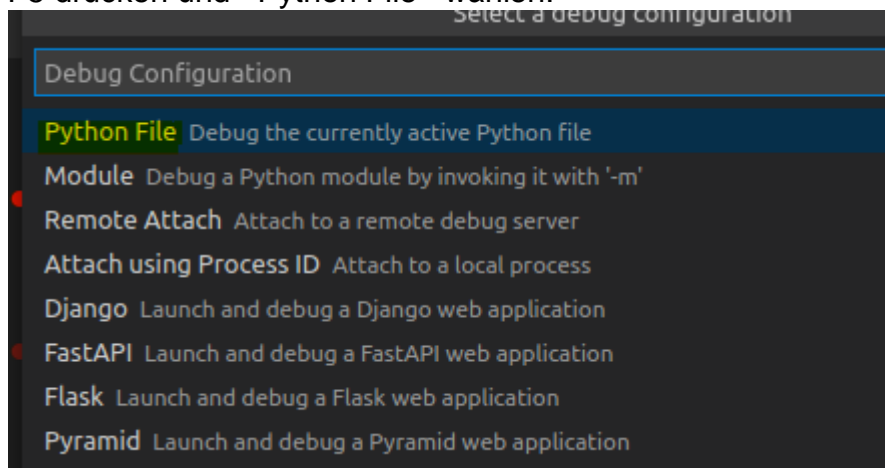
## 3.3 Debug Python Code

1. Setzen eines Breakpoint mit F9 drücken mit Cursor auf Linie 3 oder durch Maus Click auf der linken Seite der Liniennummer:

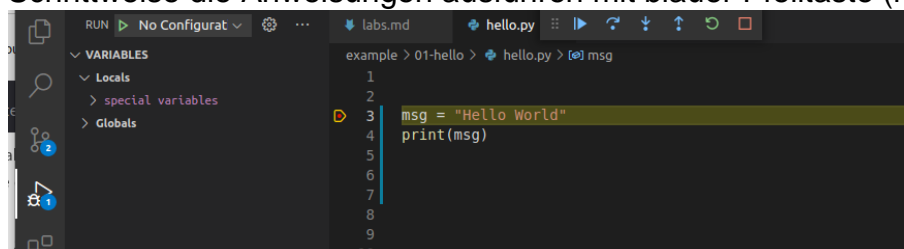
```

example > 01-hello > hello.py > ...
1
2
3 msg = "Hello World"
4 print(msg)
5
6
7
8
9
10
    
```

2. F5 drücken und «Python File» wählen.



3. Schrittweise die Anweisungen ausführen mit blauer Pfeiltaste (nach rechts):





## 3.4 Installation und Verwendung weiterer Python Bibliotheken

1. Erstellen Sie eine neue Datei **helloplot.py** mit folgendem Inhalt:

```
import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(0, 20, 100)    # Create a list of evenly-spaced numbers over
the range

plt.plot(x, np.sin(x))        # Plot the sine of each x point
plt.show()                    # Display the plot
```

2. Wenn Sie den Code das erste mal Ausführen, erhalten Sie eine Fehlermeldung:

```
Traceback (most recent call last):
  File "/home/lab/workspace/python/example/01-hello/helloplot.py", line 1,
in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
```

3. Im Terminal, wo die Fehlermeldung angezeigt wird, kann jetzt die fehlende Bibliothek installiert werden:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Python
Hello World
(base) lab@ubuntu:~/workspace/python$ /home/lab/miniconda3/bin/python /home/lab/workspace/python/example/01-hello/hello.py
Hello World
(base) lab@ubuntu:~/workspace/python$ /home/lab/miniconda3/bin/python /home/lab/workspace/python/example/01-hello/helloplot.py
Traceback (most recent call last):
  File "/home/lab/workspace/python/example/01-hello/helloplot.py", line 1, in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
(base) lab@ubuntu:~/workspace/python$
(base) lab@ubuntu:~/workspace/python$
(base) lab@ubuntu:~/workspace/python$ ls
example lab LICENSE python.iml README.md setup.md
(base) lab@ubuntu:~/workspace/python$
```

Don't use with Anaconda distributions because they include matplotlib already.

**# macOS**

**python3 -m pip install matplotlib**

**# Windows (may require elevation)**

**python -m pip install matplotlib**

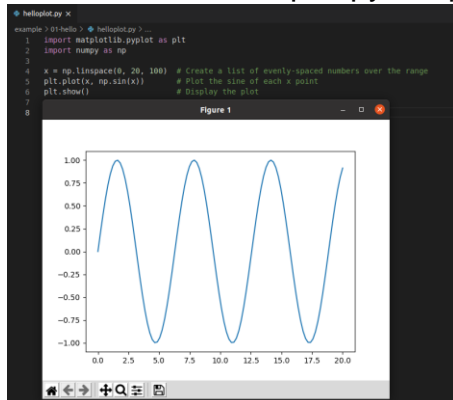
**# Linux (Debian)**

**sudo apt install python3-tk**

**python3 -m pip install matplotlib**



- Jetzt kann das helloplot.py Script nochmals ausgeführt werden:



## 3.5 Aktivierung von Unit Testing im VS Code

Das Testen in Python ist standardmäßig deaktiviert. Um Tests zu aktivieren, verwenden Sie den Befehl "Python: Configure Tests" in der Befehlspalette. Mit diesem Befehl werden Sie aufgefordert, ein Test-Framework, den Ordner mit den Tests und das Muster zur Identifizierung der Testdateien auszuwählen.

- Shift+Ctrl+P
- Eingabe: **Python: Configure Tests**
- Auswahl: **unittest:**

```

04-classes
1  in
2  f
3  pytest pytest framework
4  http://docs.pytest.org/
5  nose nose framework
6  https://nose.readthedocs.io/
7
8  # For the current case, the constructor would not be needed)
9  def __init__(self, *args, **kwargs):
10     super(MathUtilTest, self).__init__(*args, **kwargs)
11
12     def test_max(self):
13         expected = 20
14         actual = MathUtil.max(5,12,20)
15         self.assertEqual(expected, actual)
16         self.assertEqual(49, MathUtil.max(49,3,14))
17
18     def test_min(self):
19         self.assertEqual(5, MathUtil.min(5,12,20))
20         self.assertEqual(3, MathUtil.min(49,3,14))
21
22

```

- Gewünschter Ordner wählen
- Pattern für Dateinamen wählen: **\*test.py**

```

04-classes
1  in
2  f
3  test*.py Python Files beginning with 'test'
4  test_*.py Python Files beginning with 'test_'
5  *test.py Python Files containing the word 'test'
6
7  # constructor if some fields are used.

```



## 4 Ubuntu Setup

### 4.1 Miniconda

Um Miniconda unter Ubuntu 20.04 von der Kommandozeile aus zu installieren, sind nur 3 Schritte notwendig.

1. **Download aktuelles Miniconda Script**

wget [https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\\_64.sh](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh)

2. **Script für die Ausführung vorbereiten**

chmod +x Miniconda3-latest-Linux-x86\_64.sh

3. **Installation durchführen**

./Miniconda3-latest-Linux-x86\_64.sh

```
Miniconda3 will now be installed into this location:
```

```
/home/lab/miniconda3
```

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

```
[/home/lab/miniconda3] >>>  
PREFIX=/home/lab/miniconda3  
Unpacking payload ...
```

```
Preparing transaction: done  
Executing transaction: done  
installation finished.  
Do you wish the installer to initialize Miniconda3  
by running conda init? [yes|no]  
[no] >>> yes
```

```
==> For changes to take effect, close and re-open your current shell.<==
```

```
If you'd prefer that conda's base environment not be activated on startup,  
set the auto_activate_base parameter to false:
```

```
conda config --set auto_activate_base false
```

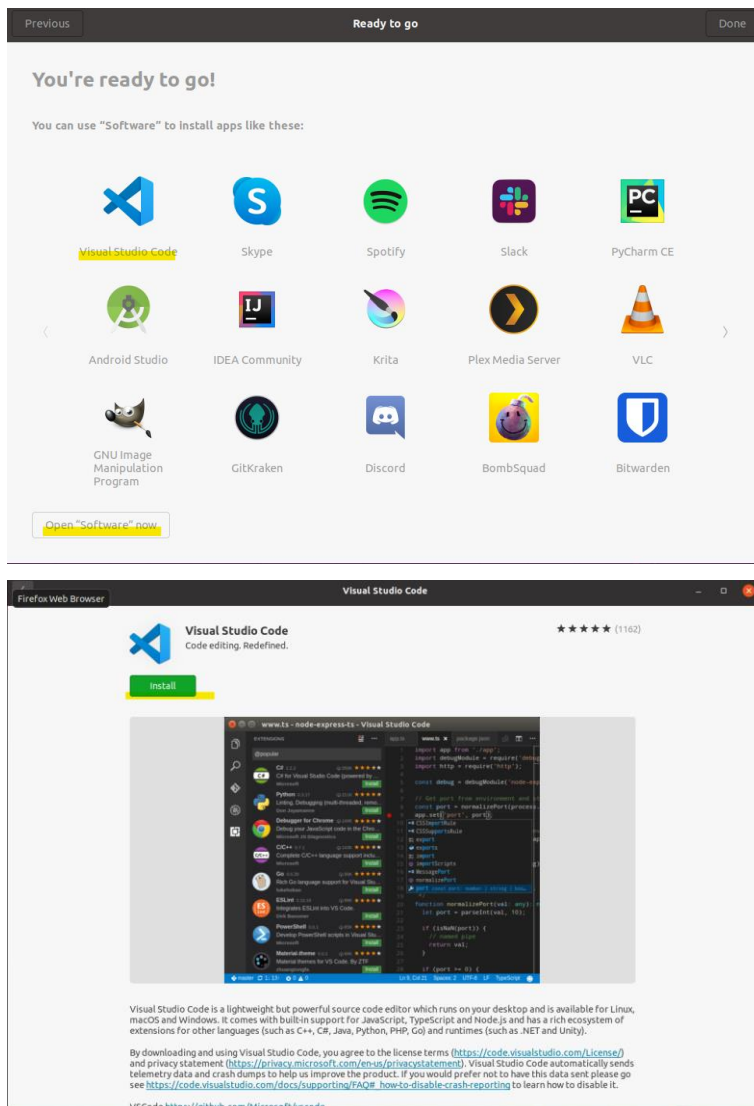
```
Thank you for installing Miniconda3!
```



## Miniconda Befehle

Command	Description
conda --version	Show version
conda info --envs	Show environments
conda create -n myenv python=3.6.2	Create environment with given python version
conda env remove -n myenv	Remove environment
conda activate myenv	Activate the myenv environment
conda list	Show packages of active environment
conda install [package name]	Install Package
conda update [package name]	Update Package

## 4.2 Visual Studio Code



Für die Konfiguration von VS Code, siehe Kapitel 2.



## 4.3 GIT

- **Installation GIT**  
sudo apt install git

```
(lab) lab@ubuntu:~/workspace$ sudo apt install git
[sudo] password for lab:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,464 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

- **Check Version**  
git --version
- **Global Einstellungen**  
git config --global user.name "Pipi Langstrumpf"  
git config --global user.email [pipi@villakunterbunt.org](mailto:pipi@villakunterbunt.org)

## 4.4 Course Beispiele

1. **Workspace Verzeichnis erstellen**  
mkdir workspace
2. **Clone Python Project**  
cd workspace  
git clone <https://github.com/iten-engineering/python>

## 4.5 Project Setup

Siehe Kapitel 2.





## 5 Anhang

### 5.1 VS Code – Tipps

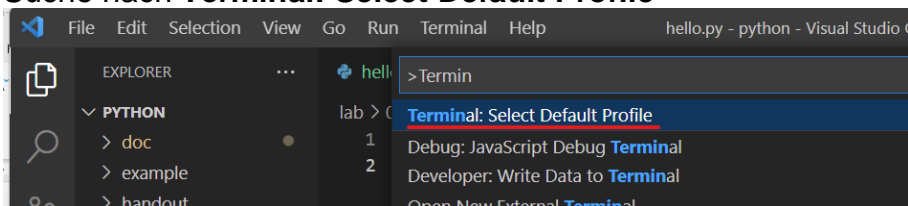
Zoom:

- Menu: View / Appearance / Zoom-In      CTRL + NUM\_PAD\_ADD
- Menu: View / Appearance / Zoom-Out      CTRL + -

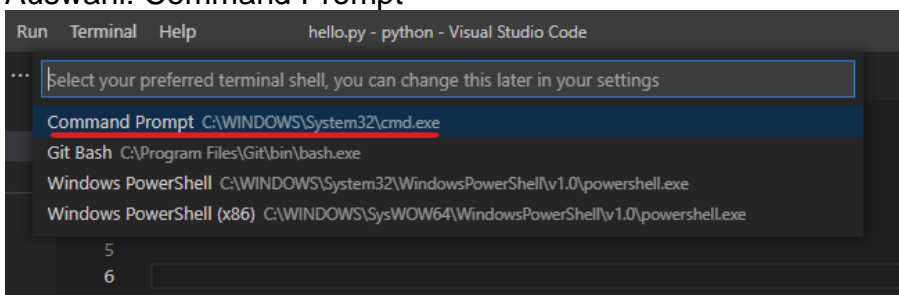
### 5.2 VS Code – Terminal Settings

1) Öffnen der Kommando Eingabe mit: **CTRL-SHIFT-P**

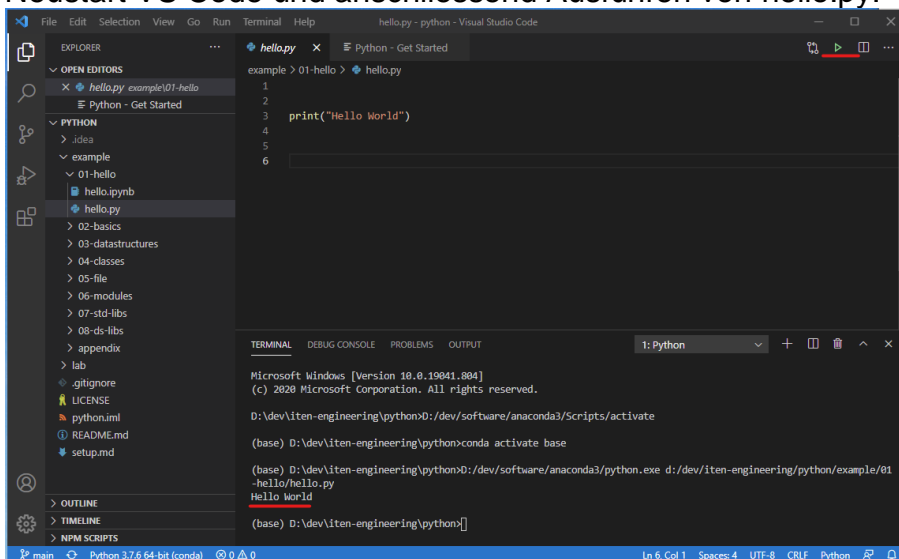
2) Suche nach **Terminal: Select Default Profile**



3) Auswahl: Command Prompt



4) Neustart VS Code und anschliessend Ausführen von hello.py:



Details siehe: <https://stackoverflow.com/questions/54828713/working-with-anaconda-in-visual-studio-code>